JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Effective Pre-rating Method Based on Users' Dichotomous Preferences and Average Ratings Fusion for Recommender Systems

Shulin Cheng*, Wanyan Wang**, Shan Yang**, and Xiufang Cheng**

Abstract

With an increase in the scale of recommender systems, users' rating data tend to be extremely sparse. Some methods have been utilized to alleviate this problem; nevertheless, it has not been satisfactorily solved yet. Therefore, we propose an effective pre-rating method based on users' dichotomous preferences and average ratings fusion. First, based on a user–item ratings matrix, a new user-item preference matrix was constructed to analyze and model user preferences. The items were then divided into two categories based on a parameterized dynamic threshold. The missing ratings for items that the user was not interested in were directly filled with the lowest user rating; otherwise, fusion ratings were utilized to fill the missing ratings. Further, an optimized parameter λ was introduced to adjust their weights. Finally, we verified our method on a standard dataset. The experimental results show that our method can effectively reduce the prediction error and improve the recommendation quality. As for its application, our method is effective, but not complicated.

Keywords

Collaborative Filtering, Data Sparsity, Fusion Filling, Preference Matrix, Recommender System

1. Introduction

The rapid development of the Internet and e-commerce applications has led to an exponential increase in information. Finding relevant and appropriate information from large-scale datasets is often a time-consuming process. Moreover, users spend a significant amount of time and energy in selecting information and products related to their interests. Recommender systems (RSs)—rapidly developing as a powerful technology in e-commerce and business analytics applications [1,2]—help users find the desired content by providing personalized products and services, to improve their information acquisition efficiency and effectively alleviate the information overload problem. The RS was first proposed by Resnick and Varian [3] in 1997. Since then, it has been widely developed, and has gradually become an important research field. Furthermore, RSs can improve users' decision quality and reduce their decision effort. Currently, RS is a major topic in the field of computer science, considerable effort has been expended by experts and scholars in the last two decades to further advance these systems.

RSs based on ratings are now widely used, where the basic idea is to use the users' interaction histories and rating information to predict unknown ratings, and to generate a recommendation list for target users to find related items they may be interested in. Historical information can be inferred from user profiles, and obtained from users' explicit consumption behaviors. A recommendation algorithm is at the core of an RS. An appropriate recommendation algorithm can significantly improve the quality of RSs, thereby enhancing the user satisfaction. Among the various methods used in RSs, the collaborative filtering (CF) method has shown superior results in research and practice. As shown in Fig. 1, the CF method uses the common rating information of users/items to calculate the similarity between users/items, and recommends new items to target users. Assume that there are two users (*A* and *B*) and three products (*a*, *b*, and *c*). If *A* and *B* have consumption behaviors on *a* and *b* simultaneously, we assume that *A* and *B* have similar interests and preferences. Therefore, when *A* has consumption behaviors on *c*, we can recommend *c* to *B*.
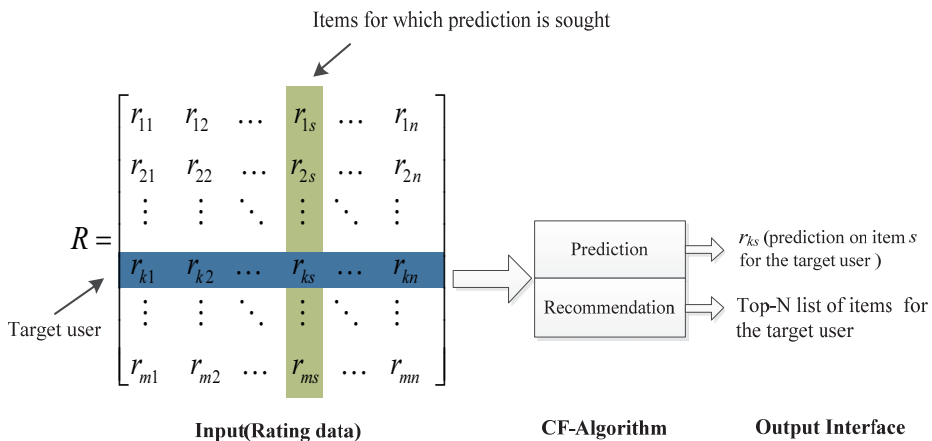

Fig. 1. Schematic of CF method.

In the field of personalized recommendation, most studies are devoted to improving the accuracy of predictions. Nonetheless, data sparsity remains the primary impediment that adversely affects the performance of RSs. Current research shows that, in large e-business systems, users generally purchase only 1% of the total number of items, and even fewer items are rated by more than two users, leading to a scenario with insufficient valuable and significant transactions for inferring reliable user similarities and targeting relevant items [4]. It is difficult to find users similar to a target user, who has rated items without sufficient ratings, making relevant recommendations harder for him/her.

In the early days, the most primitive solution to the sparsity problem was to fill in the missing data with zero values. Later, using the existing ratings' mean, median, and mode to represent the missing data, or even the imputation methods based on singular value decomposition (SVD), was proposed. However, these data filling methods cause the ratings of most vacant positions to be the same, leading to larger prediction errors and poor recommendations. They also ignore the differences between missing values, as the items should have different values based on the users' interest. Therefore, we propose a new and effective pre-rating method based on the users' dichotomous preferences and average ratings fusion for a RS. We utilize the label information of items and the existing user ratings to model and analyze their

interests and preferences. Based on this information, the items are divided into two categories. The missing ratings of items a user is not interested in are directly filled with his/her lowest rating; otherwise, the missing ratings are filled using the average ratings fusion method.

The findings of this study will contribute to the development of effective systems that can significantly reduce the sparsity hurdle, and thus, further enhance the accuracy and efficiency of the recommendations.

2. Related Work

In general, the existing CF methods are divided into model-based and memory-based methods. Model-based CF methods use machine learning algorithms to find an interaction model between users and items, and then, identify specific patterns in the data. Memory-based CF methods use the preferences of a set of like-minded users to generate recommendations that target users may be interested in. A memory-based CF method is also known as a neighborhood-based CF method. It is the most basic algorithm in RSs that has been rigorously studied in the academic field and is widely utilized in industry. Neighborhood-based algorithms are divided into two categories: user-based and item-based CF algorithms. Neighborhood-based algorithms can determine data linkages better than model-based algorithms, and may recommend items significantly different from the users' previous interests and preferences, thereby improving the novelty and surprise of recommendations, while avoiding the singularization of their interests and preferences.

A user-based CF algorithm is the most frequently used algorithm in RSs, and its appearance marks the beginning of RSs. The user-based CF framework was proposed by Goldberg et al. [5]. At the core of the user-based CF recommendation method is the assumption that users with similar interests in the past will have common interests in the future, which can be utilized to predict a target user's rating for an item through a weighted combination of selected neighboring users' ratings. Therefore, when user A needs a personalized recommendation, the RSs can first find users with similar interests as A, and then, recommend products that these users like, which A has not yet consumed. However, a user-based CF algorithm is not very scalable when facing tens of thousands of potential neighbors. Therefore, an item-based CF method was proposed. An item-based method assumes that the target user may like items similar to their favorite items [6,7]. The CF algorithm generates recommendations based on similarity rules. A user-based CF algorithm recommends items favored by other users who are highly similar to the target user, while an item-based CF algorithm finds items highly similar to the purchased items from a list of items the target user has not purchased.

However, the problem of data sparsity significantly limits the applicability of a neighborhood-based CF algorithm, thereby reducing the recommendation algorithm's performance. To solve this problem, several scholars have proposed different methods in related studies and made significant progress. Factorization is regarded as an effective method for solving the problem of data sparsity, and is widely used in RSs [8]. To address the problem of data sparsity, Guo et al. [9] proposed a conceptual model for RSs using a new information source—prior ratings—by integrating the presence of environmental factors whose effects on product evaluation was not studied previously. Hu et al. [10] proposed a hybrid personalized random walk algorithm to handle the data sparsity in web service recommendations based on an existing random walk algorithm and the characteristics of web services. A personalized random walk algorithm was implemented on the service graph to identify more similar neighbors for the current

target user requesting the Web service and the target users of each candidate web service. Kermany and Alizadeh [11] proposed a new hybrid method based on an enhanced fuzzy multi-criteria CF, which combines demographic information and an item-based ontology semantic filtering method to effectively alleviate the sparsity problem and improve the recommendation accuracy.

3. Pre-rating Strategies and Methods

With the increasing popularity of the Internet, the scale of RSs continues to expand, and the user-item rating matrix has become extremely sparse. This reduces the accuracy of the similarity measurement method in the traditional CF recommendation algorithm, thereby reducing the RS quality. To overcome the challenges caused by sparsity, we propose a new pre-rating data filling method. First, the items are divided into two categories: user-interested and user-disinterested; subsequently, different methods are used to predict and fill the user ratings.

3.1 Preferences Classification Based on Dichotomy Strategy

In RSs, users have different interests and preferences for different items. We often find that users are interested in items with specific labels, and not interested in any item with other labels. In many cases, the item labels may attract users. The label can be expressed in the form of an item type, such as a movie type, book type, and music style. Considering movie recommendations as an example, one movie can be a romantic-comedy, while another one can be a suspense-thriller. Therefore, users interested in romantic movies would like the former to be recommended to them. Thus, we can adopt a dichotomy strategy based on a threshold to model the users' interest preferences, using the type labels of items to classify them into two categories: items a user is interested in and items a user is not interested in. A flowchart of the proposed method is shown in Fig. 2.
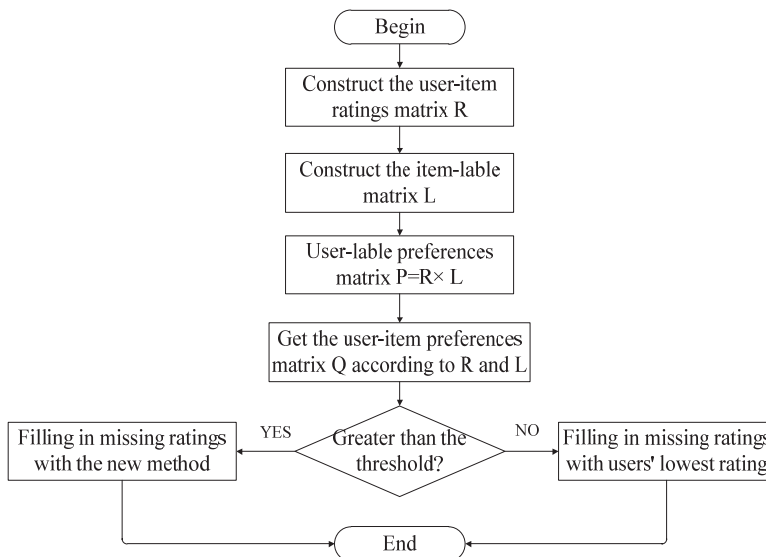

Fig. 2. Flowchart of the proposed method.

First, the user–item rating matrix R is constructed according to the rating information of users on the items in the dataset. Here, $U=\{U_1,U_2,U_3,\ldots,U_m\}$ is the user set composed of m users, $I=\{I_1,I_2,I_3,\ldots,I_n\}$ is the item set composed of n items, and r_{ks} represents the k-th user's opinion on the s-th item. The level of the ratings implies the users' degree of interest in the corresponding items. When the rating is higher, the user's interest in the item is higher, and vice versa.

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1s} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2s} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{k1} & r_{k2} & \cdots & r_{ks} & \cdots & r_{kn} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{ms} & \cdots & r_{mn} \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

Each item may have multiple labels. The label matrix for a collection of n items and l labels can be expressed as a $n{\times}l$ matrix L. Here, 1 implies the item has a label, and 0 implies the item does not have this label. The user preference matrix P for item labels is constructed from the ratings matrix R and labels matrix L, as follows.

$$P = R \times L \tag{1}$$

Then, the user's interest and preference for the items are calculated as

$$q_{ks} = \sum_{i \in l} p_{ki}, \quad if \ r_{ks} \neq 0 \ and \ L_{si} = 1 \tag{2}$$

where q_{ks} represents the k-th user's preference for the s-th item, and p_{ki} represents the k-th user's preference for the i-th item. $L_{si}=1$ implies the s-th item contains the i-th label; that is, if the item contains multiple labels, the user's preference for the item is approximately the cumulative value of his/her preference for the included labels. Subsequently, the user–item preference matrix Q can be obtained as follows.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1l} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2l} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & p_{k3} & \cdots & p_{kl} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots & p_{ml} \end{bmatrix} \qquad Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1s} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2s} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ q_{k1} & q_{k2} & \cdots & q_{ks} & \cdots & q_{kn} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{ms} & \cdots & q_{mn} \end{bmatrix}$$

After obtaining the user–item preference matrix Q, we analyze and judge the user's interests and preferences. First, we construct Eq. (3) to calculate the threshold.

$$\theta_u = \alpha * \frac{\sum_{i=1}^{n} q_{ui}}{n} \tag{3}$$

where θ_u represents the preference threshold of the u-th user. Each user in the user set has a specific threshold. α is a parameter used to adjust the threshold (the details will be discussed in Subsection 4.2). Then, we judge a user's interest level according to the indicator function in Eq. (4).

$$S_{ui} = f(\alpha) = \begin{cases} 1, & q_{ui} \geq \theta_u \\ 0, & q_{ui} < \theta_u \end{cases} \tag{4}$$

If the u-th user's preference for the i-th item is greater than his/her preference threshold, then the user is considered to be interested in the item, which is marked as $S_u = 1$; otherwise, the user is considered to be not interested in the item, which is marked as $S_u = 0$.

3.2 Pre-rating Method

Based on the division of items into two categories, in this section, we will use different methods for pre-rating different items to fill in the missing data.

The missing ratings for items of no interest to the users are directly filled with their lowest ratings. Because different users have different rating scales, their ratings for items they are not interested in are also different.

$$\hat{r}_{ui} = \min \ \{r_{u1}, \ r_{u2}, \ \cdots, \ r_{us}, \ \cdots, \ r_{un}\} \tag{5}$$

The users' rating characteristics and the attributes of their items of interest are comprehensively considered, and a filling method that combines the users' and items' average ratings is adopted; then, their respective weights are adjusted by introducing a related parameter.

There will be differences in the rating habits of each user in the user set. These differences embody the users' rating biases. Some users have a higher requirement for movies; they are relatively stricter with their scoring and may generally give a lower rating to movies, while others will give a higher rating as long as the movie is not terrible. Therefore, to minimize the influence of user rating biases on items, we utilize the average rating using Eq. (6).

$$\overline{r_u} \ = \ \frac{\sum_{i=1}^{n} r_{u,i}}{count\left(r_{u,i} \neq 0\right)} \ , \qquad if \ \ r_{u,i} \neq 0 \tag{6}$$

Different items in the item set have different feature labels and degrees of user preference. Some movies may be well made or involve popular actors, which will result in higher ratings than other movies. When predicting a user's rating of an item, it is necessary to consider both his/her preference and the degree of user preference for the item itself. The average rating of all users who rated the item can be used to represent the average rating of the item, and can be calculated as

$$\overline{r_i} \ = \ \frac{\sum_{u=1}^{m} r_{u,i}}{count\left(r_{u,i} \neq 0\right)} \ , \qquad if \ \ r_{u,i} \neq 0 \tag{7}$$

Then, if the u-th user's rating for the i-th item in the rating matrix is missing, it can be predicted and filled in using Eq. (8).

$$r_{ui}' = \lambda \overline{r_u} + (1 - \lambda) \overline{r_i} \qquad (8)$$

where λ is a parameter that adjusts the weight of average ratings to predict the missing ratings. After filling in the missing data with the above method, the ratings matrix is saturated. In other words, for any item belonging to I, the rating of u for i is shown in Eq. (9).

$$r_{ui} = \begin{cases} r_{ui}, & r_{u,i} \neq 0 \\ \hat{r}_{ui}, & r_{u,i} = 0 \ \ and \ \ S_{ui} = 0 \\ r_{ui}', & r_{u,i} = 0 \ \ and \ \ S_{ui} = 1 \end{cases} \qquad (9)$$

4. Experiments and Analysis

4.1 Experiment Design

To verify the effectiveness of our proposed method, we conducted several experiments on the MovieLens dataset (https://grouplens.org/datasets/movielens/). MovieLens is a general dataset for RSs, containing the rating and attribute information (the type label of a movie) of items. We chose the 100k dataset, which contains 100,000 ratings from 943 users on 1,682 movies, and a high data sparsity of 93.70%.

We used MATLAB 2018a to perform the experiments on a computer with a quad-core 2.80 GHz processor and Windows 10 operating system. We randomly selected 20% of the user ratings as the test set, and the rest of the dataset was used as the training set. The experiments were performed four times, and the results were averaged to reduce the influence of a few extreme values.

The cosine function was used to calculate the similarity between users and predict the ratings, as follows.

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui} * r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} * \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \qquad (10)$$

where u and v represent two users in the user set, and $Sim(u, v)$ represents the similarity between them. I_{uv} represents a collection of items rated by both u and v. r_{ui} and r_{vi} represent the ratings of u and v for item I, respectively. After obtaining the similarity matrix, Top-N users with the highest similarity are selected as the neighbors of the target user, and the target user' rating for item I is predicted based on the ratings of the neighboring users, which is calculated as follows.

$$\tilde{r}_{uj} = \frac{\sum_{v \in N_u} r_{vj} * sim(u,v)}{\sum_{v \in N_u} sim(u,v)} \qquad (11)$$

In this study, we used mean absolute error (MAE) and root-mean-square error (RMSE) as the evaluation metrics for our proposed method. When the error is greater, the accuracy is lower, and vice versa. The calculation formulas for MAE and RMSE are shown in Eq. (12) and Eq. (13), respectively.

$$MAE = \frac{\sum_{i=1}^{T} |\tilde{r}_{ui} - r_{ui}|}{T} \qquad (12)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T}(\tilde{r}_{ui} - r_{ui})^2}{T}} \tag{13}$$

where T represents the number of predicted ratings, r_{ui} denotes the user's actual rating for the item, and $\tilde{r}_{ui}$ represents the user's predicted rating for the item.

4.2 Experiment (1): Sensitivity Analysis of Parameters

Three parameters (α, λ, and N) are used in the proposed method. α is used as a parameter to determine the threshold value. λ is used to control the weight of average ratings. N is the size of the user's neighborhood. Several experiments were conducted to demonstrate the effects of different parameter values on our proposed method's performance.

A reasonable threshold is vital for judging the users' interests and preferences, and helps determine the method to be used to fill in the missing data. To analyze the influence of the threshold parameter α on the proposed method, we conducted some related experiments with $\lambda = 0.5$ and $N = 5$. When $\lambda = 0.5$, the influence of the users' and items' average ratings is excluded. The influence of N on the proposed method is discussed later. Here, N is the minimum value of the nearest neighbor range. Fig. 3 shows that both MAE and RMSE decrease with a decrease in the threshold parameter α. When α is smaller, more data is filled using the proposed method, and less data is filled with users' lowest scores, which is consistent with the fact that there are only a few items of no interest to the user.
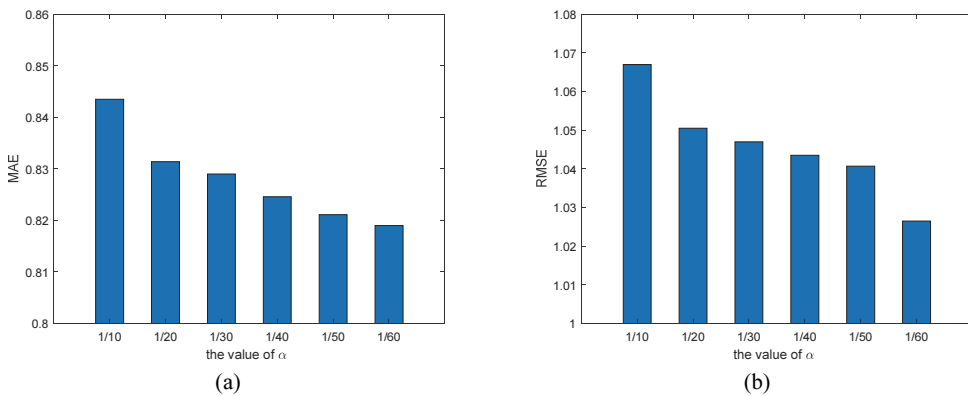

Fig. 3. Effect of parameter (α) on the performance metrics: (a) MAE and (b) RMSE.

In the proposed algorithm, the weight parameter λ is used to adjust the weight of the average rating of users and items. The related experiments were performed with $N = 5$. Fig. 4 shows how different values of λ affect the performance metrics. The value of λ ranged from 0.1–0.9 with a step of 0.1. As can be seen from these results, the MAE and RMSE measurements increase with an increase in λ. Thus, increasing the value of λ has a negative effect on the MAE and RMSE. In the MovieLens 100k dataset, the best experimental results were obtained with $\lambda = 0.1$. At this time, the values of MAE and RMSE were 0.7880 and 0.9939, respectively.
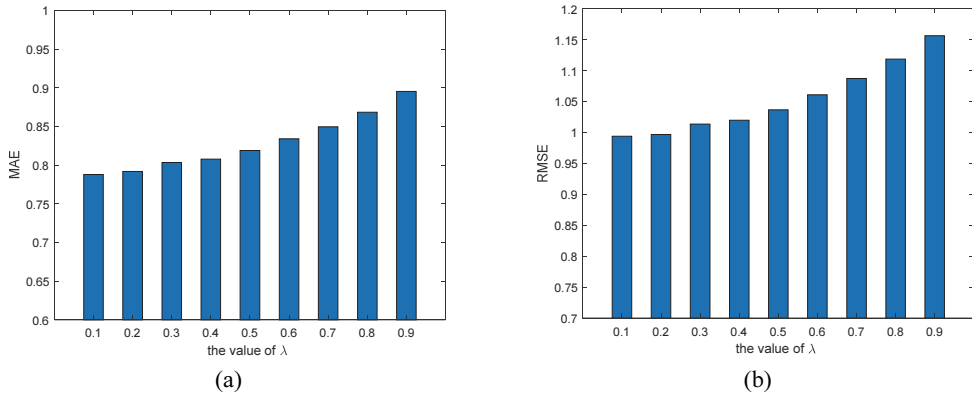
Fig. 4. Effect of parameter (λ) on the performance metrics: (a) MAE and (b) RMSE.

Finally, the experiments were repeated with different values of N to evaluate the proposed method's performance; the results are shown in Fig. 5. The range of N was set between 5–50 with a step of 5. As can be seen from these results, when the neighborhood size changes, both MAE and RMSE tend to stabilize. In other words, our method is less affected by the neighborhood size.
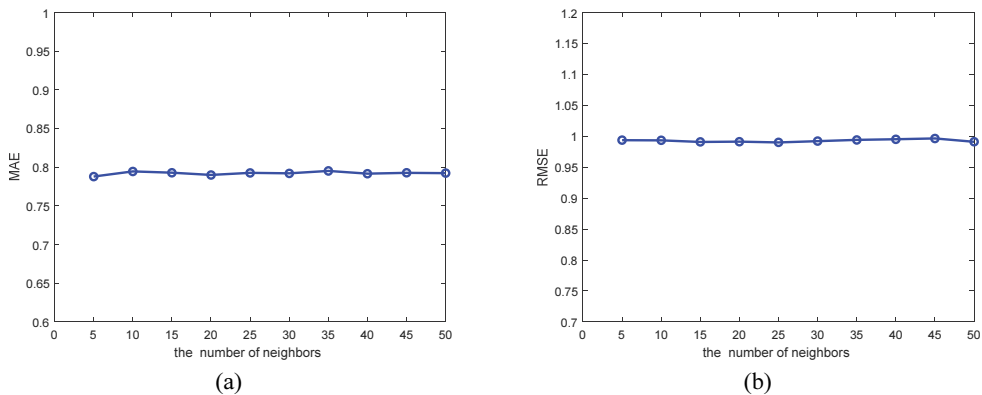

Fig. 5. Effect of parameter (N) on the performance metrics: (a) MAE and (b) RMSE.

4.3. Experiment (2): Comparison with Other Methods

In this section, the performance of the proposed method is compared with that of other methods, while handling data sparsity. Table 1 shows the results of the experiments on the MovieLens dataset. Among these, "Zero-rating" represents the filling of missing data with 0; "Avg-rating" denotes the filling of missing data with the average of existing ratings; "Med-rating" represents the filling of missing data with the median of existing ratings; and SVD is a traditional matrix decomposition method [12]. Our method obtains the best MAE and RMSE values when compared to the other methods on this dataset. Our proposed method improves the accuracy by approximately 3.28% and 2.91%, respectively, on the MAE and RMSE of SVD—i.e., the second-top performer. Thus, the proposed rating prediction method can effectively reduce the prediction error, thereby improving the recommendation quality.

Table 1. Comparisons with other methods

Methods	MAE	RMSE
Zero-rating	1.6560	2.0354
Avg-rating	1.1743	1.5126
Med-rating	0.9165	1.1394
SVD	0.8147	1.0237
Our-method	0.7880	0.9939

5. Conclusion

Focusing on the problem of data sparsity in a real rating dataset and motivated by the consideration of rating differences between items with missing values for users, we proposed a users' preference dichotomy strategy and pre-rating method that fuses the average ratings of users and items. To differentiate the items with missing values, some of which may or may not be of interest to the users, we constructed a new user-item preference matrix to analyze and model user interests and preferences using an initial user-item rating matrix. Then, a threshold parameter was introduced to divide the items into two categories: those that are of interest and those that are not of interest. For both categories of items, we filled the missing values with different strategies. Based on these strategies, we proposed a method that fuses the average ratings of users and items. Finally, a corresponding parameter was introduced and optimized to adjust the weight of these average ratings. The experimental results show that our proposed method can effectively reduce the prediction error and improve the recommendation quality, demonstrating the significance of different filling strategies.

Acknowledgement

References

[1] A. Montarnal, W. Mu, F. Benaben, J. Lamothe, M. Lauras, and N. Salatge, "Automated deduction of cross-organizational collaborative business processes," *Information Sciences*, vol. 453, pp. 30-49, 2018.

[2] P. Chamoso, A. Rivas, S. Rodriguez, and J. Bajo, "Relationship recommender system in a business and employment-oriented social network," *Information Sciences*, vol. 433, pp. 204-220, 2018.

[3] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997.

[4] N. Idrissi and A. Zellou, "A systematic literature review of sparsity issues in recommender systems," *Social Network Analysis and Mining*, vol. 10, no. 1, article no. 15, 2020. https://doi.org/10.1007/s13278-020-0626-2

[5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61-70, 1992.

[6] C. Li and K. He, "CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, article no e4092, 2017. https://doi.org/10.1002/cpe.4092

[7] E. Shmueli and T. Tassa, "Secure multi-party protocols for item-based collaborative filtering," in *Proceedings of the 11th ACM Conference on Recommender Systems*, Como, Italy, 2017, pp. 89-97.

[8] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6287-6304, 2017.

[9] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Leveraging prior ratings for recommender systems in e-commerce," *Electronic Commerce Research and Applications*, vol. 13, no. 6, pp. 440-455, 2014.

[10] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 782-794, 2014.

[11] N. R. Kermany and S. H. Alizadeh, "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques," *Electronic Commerce Research and Applications*, vol. 21, pp. 50-64, 2017.

[12] C. Feng, J. Liang, P. Song, and Z. Wang, "A fusion collaborative filtering method for sparse data in recommender systems," *Information Sciences*, vol. 521, pp. 365-379, 2020.

Shulin Cheng https://orcid.org/0000-0002-3744-6335

He received the M.Sc. degree in mechatronic engineering from Guandong University of Technology, in 2004, and the Ph.D. degree in computer application technology from Shanghai University, in 2018. He is currently a Professor in the School of Computer and Information, Anqing Normal University. His current research interests include intelligent information processing, personalized recommendation, social networks application, data mining, and machine learning.

Wanyan Wang https://orcid.org/0000-0003-4848-6989

She received B.S. degree in information management and information system from Anqing Normal University in 2018. She is currently studying for the M.S. degree in statistics from Anqing Normal University. Her research interests include intelligent information processing, data mining, and personalized recommendation.
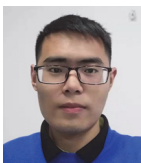
Shan Yang https://orcid.org/0000-0001-7750-3579

He received B.S. degree in information management and information system from Anqing Normal University in 2019. He is currently studying for the M.S. degree in software engineering from Anqing Normal University. His research interests include intelligent information processing, social networks application and data mining.

Xiufang Cheng https://orcid.org/0000-0001-6780-4730
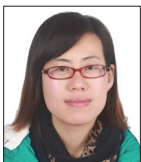
She received the M.Sc. degree in computer application technology from Army Officer University, in 2017. She is currently a lecture in the School of Computer and Information, Anqing Normal University. Her current research interests include expert system, personalized recommendation.